NISTIR 89-4139
(Volume 1)

A DETAILED DESCRIPTION OF THE KNOWLEDGE-BASED SYSTEM FOR PHYSICAL DATABASE DESIGN - VOLUME I

Christopher E. Dabrowski

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards
and Technology
National Computer Systems Laboratory
Information Systems Engineering Division
Gaithersburg, MD 20899

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary

NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
Raymond G. Kammer, Acting Director

NIST

NISTIR 89-4139
(Volume 1)

A DETAILED DESCRIPTION OF THE KNOWLEDGE-BASED SYSTEM FOR PHYSICAL DATABASE DESIGN - VOLUME I

Christopher E. Dabrowski

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology
National Computer Systems Laboratory
Information Systems Engineering Division
Gaithersburg, MD 20899

August 1989

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
Raymond G. Kammer, Acting Director

A DETAILED DESCRIPTION OF THE KNOWLEDGE-BASED SYSTEM FOR PHYSICAL DATABASE DESIGN

VOLUME I

Christopher E. Dabrowski

National Computer Systems Laboratory
National Institute of Standards and Technology

ABSTRACT

A knowledge-based system for physical database design has been developed at the National Computer Systems Laboratory. This system was previously described in NIST Special Publication 500-151. This is a follow-up report to that publication which describes the knowledge bases of this system in detail. The description includes a complete explanation of each component of the knowledge base together with the actual rules used by the system. There are two volumes to this report. Volume I contains explanatory text describing each knowledge base. Volume II contains the actual rules.

Key words: certainty factor; entity-relationship model; inference engine; knowledge-based system; logical data structure; physical database design.

TABLE OF CONTENTS

1. INTRODUCTION: THE PURPOSE OF THIS REPORT

The purpose of this report is to provide a detailed description of knowledge bases in the Knowledge-Based System (KBS) for Physical Database Design, described in NBS Special Publication 500-151. This publication discussed the background, purpose, and problem solving architecture of the knowledge-based system developed at the National Institute of Standards and Technology (formerly the National Bureau of Standards). The function of each knowledge base was also discussed in SP 500-151, but specific rules were not mentioned. This report has been compiled because these rules, together with a detailed description of the problem solving knowledge of the KBS, may be of value to future researchers.

Specifically, the information may benefit researchers interested in development of automated systems for physical database design, with the heuristics from the KBS knowledge bases serving as a basis for more powerful and more specialized systems. See [STOR88] for a survey of research into expert systems for database design. Unlike many other systems of this type, the KBS contains actual heuristics obtained from domain experts which can be useful in other efforts. This work may also be of value as an example case study of a knowledge base, to be used by researchers interested in expert systems and knowledge organization.

This report does not describe how to use the KBS. Nor does it describe input parameters to the system. This information is more appropriate for a user's manual, which is yet to be written.

1.1 Necessary Background To Understand This Report

A few prerequisites are probably needed to better understand the contents of this report. The reader should have a working knowledge of the Entity Relationship Attribute Model [CHEN76] and be familiar with the general concepts of physical database design. Preferably, the reader will be familiar with the research on physical database design of [CARL80], [MARC78], etc. since this is the basis for the KBS. The reader should understand the contents of Special Publication 500-151. Also, familiarity with rule-based deduction and with unification-based pattern matching [NILS80] is desirable, but not necessary. See also [CLOC84] or [CUGI87].

Although this report assumes the reader is familiar with the above, many of the key concepts and definitions will be restated in this report for the reader's convenience and to provide a better understanding of individual rules. In a couple of instances, discussions found in SP 500-151 are also restated and expanded on. A description of backward chaining and pattern matching as it applies to rules in the KBS is also provided.

1.2 The Organization Of This Report

The report is organized as follows. Chapter 2 provides a general discussion about the knowledge bases and rules in the KBS. Chapters 3-7 contain individual descriptions of the five knowledge bases. Information is provided to the reader in two different forms: descriptive text about the content of each knowledge base and the actual rules which make up the knowledge base. Correspondingly, there are two parts to the report: Volume I contains descriptive text for the knowledge bases; Volume II consists of five Appendices containing the rules together with brief comments and a glossary of selected terms.

The two volumes are meant to be read in parallel. The organization of the second volume corresponds to that of the first. The recommended method of reading this report is to read a section of the text in Volume I first, followed by the corresponding section of rules in Volume II. Each of the five knowledge bases of the KBS is covered in a separate section. Within sections, individual subsections describe knowledge base subdivisions known as rule groups, which will be more precisely described below. Many rule groups are further subdivided and presented in distinct parts for better organization and clarity.

Readers may wish to refer to individual rule groups. However, since rule groups use information concluded by other rule groups, it will often be necessary to look at two or more parts of the report for a complete understanding of the contents of a rule group or even a single rule. For this reason, the text contains many cross references to other sections in this report. Relevant sections of SP 500-151 are also frequently referenced to provide further background. Since it is anticipated this report will be used for spot reference of individual rules and rule groups, definitions of terms sometimes appear in more than one place, for the convenience of the reader.

1.3 Acknowledgements

I would like to thank Dr. David K. Jefferson, who provided the motivation for the Knowledge-Based System for Physical Database Design, and who served as the principal domain expert. Dr. Jefferson and I spent many profitable hours discussing heuristics and planning the design of this system. Appreciation is also due to Dr. John V. Carlis and Dr. Salvatore T. March, both of the University of Minnesota. Their extensive research into physical database design provided a basis for our knowledge-based system, and they served as secondary domain experts.

2. GENERAL DISCUSSION OF THE KNOWLEDGE BASES OF THE SYSTEM

For the purposes of this report, a knowledge base is a collection of rules, each of which expresses some aspect of knowledge about the physical database design domain. The function of a knowledge base is to facilitate the application of knowledge in problem solving, and to organize knowledge about a domain for the benefit of researchers and domain experts.

The KBS uses knowledge bases of rules to do physical database design. The KBS has five knowledge bases:

* The Control Module Knowledge Base (referred to as the High Level Knowledge Base in SP 500-151) manages the operation of the KBS and determines the initiation and execution of actions.

* The Entity Relationship Analysis Knowledge Base contains rules for characterizing individual entities and relationships, and for characterizing small interrelated groups of entities and relationships.

* The Representation Selection Knowledge Base contains rules for initially selecting a reasonable set of representations which may be considered during subsequent processing by the KBS, and also contains rules for direct reduction of problem size by eliminating individual representations.

* The Cluster Division Knowledge Base contains rules which determine breakpoints within the LDS for the purpose of subdividing a large problem and creating entity clusters (See Section 4.2.1 of SP 500-151).

* The Skeleton Generation Knowledge Base contains rules for identifying efficient and inefficient canonical records created by selecting representations, and for generating alternatives.

These knowledge bases are the same as those discussed in SP 500-151.

2.1 The Format Of Rules.

The format of the rules is discussed in Section 3.1 of SP 500-151. What follows is a more detailed description which should be more helpful in understanding this report.

Rules are internal data structures used to represent small pieces of knowledge about what action to take or what to conclude under a particular set of conditions. Rules have two parts: the IF part, or antecedent, lists one or more conditions; the THEN part,

or consequent, contains conclusions which are reached if the conditions in the IF part are satisfied. Individual conditions are represented in the IF part clauses. Each is a pattern to be matched against actual data. The actual data consists of a database of individual fact expressions, which may be part of the original problem statement or may have been concluded by other rules. The clauses in the IF part represent a conjunction of individual conditions; each clause must be satisfied. The following is an example rule.

```
IF   DEGREE_OF ?REL-ID ?ENT1 ?ENT2 1 M
     DEPENDENT_ON ?ENT2 ?ENT1

THEN PROPOSE_REPRESENTATION ?REL-ID ?ENT1 ABSORBS ?ENT2
```

The rule states that if the degree of relationship ?REL-ID from ?ENT1 to ?ENT2 is one to many, and if ?ENT2 is dependent on ?ENT1, then propose (e.g. suggest) absorption of ?ENT2 into ?ENT1.

2.2 Certainty Factors.

The rules in the KBS may use certainty factors. Certainty factors are a numeric measure of the degree to which a fact is believed to be true (or false) by the knowledge-based system. Absolute certainty is 1.0; absolute denial is -1.0. If a consequent is concluded by a given rule, then the certainty of that consequent may be provided by a certainty factor associated with that rule, or may be derived from the certainty factors of the facts which satisfied the antecedent portion of that rule. The inference engine is responsible for the derivation of certainty factors.

Certainty factors are useful in making judgmental conclusions, taking into account different and possibly conflicting evidence. Use of certainty factors allows the KBS to select a single design alternative from among several possible alternatives and to make "best guess" approximations of the best choice. In addition, facts which are concluded by applications of several rules may be assigned a combined certainty.

A number of algorithms exist for determination of combined certainty [THOM85]. The method used in the KBS is based on the Bernoulli formula [SHAF76]. Using this formula, if two certainty factors C_1 and C_2 are both positive or both negative and are associated with different rules concluding the same fact, then they are combined using the function $C_3 = C_1 + C_2 * (1 - C_1)$. The final positive and negative factors are combined by simple summation. Section 3.1 of SP 500-151 provides further explanation and an example of the Bernoulli formula in use.

Not every rule has an explicit certainty factor. In some cases, the strength of the conclusion of a rule is determined by a computed certainty factor derived from the individual strengths of any existing fact expressions which match the antecedent clauses in the IF part. If a rule has a computed certainty factor, the individual strengths of the matching antecedent clauses are computed on the basis of the Bernoulli rule. The Bernoulli formula is applied to fact expressions providing positive and negative certainties individually. The resulting positive and negative numbers are then added to produce a single certainty factor.

2.3 Types Of Rules In The Knowledge Bases.

The KBS contains several different types of rules.

Definition Rules. These rules contain definitions of the structural characteristics of the entity-relationship-attribute model such as entity types and relationship types. They are not heuristics. For instance, the two rules which appear below contain the definition of a one to many relationship and of entity dependency.

```
IF   ACTUAL_DEGREE ?REL_ID ?ENT1 ?ENT2 1 ?REL_CARDINALITY
     *FUNCTION_CALL* *>* ?REL_CARDINALITY 1

THEN DEGREE_OF ?REL_ID ?ENT1 ?ENT2 1 M


IF   PRIMARY_IDENTIFYING_RELATIONSHIP ?ENTITY ?REL_ID
     DEGREE_OF ?REL_ID ?ENT1 ?ENTITY 1 ?REL_CARDINALITY

THEN DEPENDENT_ON ?ENTITY ?ENT1
```

The fact expressions for relationship degree, partially identifying relationships, and entity dependency are included here without explanation to provide examples of rules. The derivation of these fact expressions will be discussed in Section 4.4.

Numeric Computation Rules. Like Definition Rules, these rules also do not represent heuristic information. Their function is to provide information obtained by numeric computation for use by the rest of the system. Usually, external function calls are used to obtain this information. The following example shows a rule to compute frequency of access to an entity of single record retrieval.

```
IF   ENTITY ?ENTNAME
     *FUNCTIONAL_CALL* *TOTAL_DIRECT_FREQUENCY* ?ENTNAME LARGE_SUBSET ?FREQUENCY

THEN ENTITY_ACTIVITY ?ENTNAME RETRIEVAL LARGE_SUBSET ?FREQUENCY
```

Characterization Rules. Rules which represent analytical knowledge play an important role in the KBS, especially in the analysis of structure and workload. Significant workload exists for entities along a relationship if a substantial amount of retrieval activity focuses on either entity by traversing the relationship. Workload complexity (Section 2.1.2 of SP 500-151) indicates that there is significant, and sometimes conflicting, activity involving two or more entities which are related to each other. The greater and more varied the activity, the greater is the level of workload complexity. Analytical conclusions about workload types and levels and resulting workload complexity, as well as analytical conclusions based on structural features of the LDS, are referred to as characterizations (Section 2.3 of SP 500-151). Workload characterizations are based on numeric data, such as frequency of activity along a particular relationship, which is obtained from fact expressions concluded by numeric computation rules.

```
IF   (RELATIONSHIP_ACTIVITY ?REL_ID ?ENT1 ?ENT2 ?TYPE SINGLE ?FREQUENCY)
     (FUNCTION_CALL *IS-ONE-OF* ?TYPE (RETRIEVAL NON_FORWARDING_ACTIVITY)
     (MINIMUM_CUTOFF_FOR RETRIEVAL ?CUTOFF_FREQ)
     (FUNCTION_CALL *>* ?FREQUENCY ?CUTOFF_FREQ)

THEN (ACTIVITY_CHARACTERIZATION_FOR ?REL_ID ?ENT1 ?ENT2 ?TYPE SINGLE SIGNIFICANT)
```

This rule states that if a relationship is traversed from entity ?ENT1 to entity ?ENT2 with frequency ?FREQUENCY, and this frequency is greater than the cutoff frequency ?CUTOFF_FREQ, then characterize this activity level as being significant.

Characterizations about workload, as well as other information, can be combined to identify more acute workload complexity problems and/or characteristics pertaining to individual entities and relationships.

For instance, in the example below, the rule concludes the existence of significant bidirectional activity by matching previously concluded information about activity in each direction along a one to many relationship. Note that the variables for subset size permit matching activity of different sizes.

```
IF   (ACTIVITY_CHARACTERIZATION_FOR ?REL_ID ?ENT1 ?ENT2 RETRIEVAL ?SUBSET_SIZE1 SIGNIFICANT)
     (DEGREE_OF ?REL_ID ?ENT1 ?ENT2 1 M)
     (ACTIVITY_CHARACTERIZATION_FOR ?REL_ID ?ENT2 ?ENT1 RETRIEVAL ?SUBSET_SIZE2 SIGNIFICANT)

THEN (RELATIONSHIP_CHARACTERIZATION ?REL_ID BIDIRECTIONAL_ACTIVITY SIGNIFICANT_WORKLOAD_COMPLEXITY)
```

A number of diverse workload characterizations involving a closely connected group of entities and relationships indicates that the group is interdependent and must be processed together by the KBS. Section 4.2 of this report provides a detailed description of system workload together with the different types of possible workload characterizations.

Heuristic Rules. Heuristic rules are heuristics expressed in rule form. In the KBS, two important functions of heuristic rules are to propose relationship representations and select breakpoints. An important feature of heuristic rules is that they infer probable solutions on the basis of information which is topically unrelated or indirectly related. For instance, in the rule below, absorption is suggested for a one to many relationship in which the "many" entity is dependent on the "1" entity.

```
IF   DEPENDENT_ON ?ENT2 ?ENT1
     DEGREE_OF ?REL-ID ?ENT1 ?ENT2 1 M

THEN PROPOSE_REPRESENTATION ?REL-ID ?ENT1 ABSORBS ?ENT2
```

The concepts of relationship degree and entity dependency are topically unrelated to the idea of absorption. However, according to the heuristic, both the existence of a one to many degree and dependency indicate that absorption is probably a good representation as is reflected in the certainty factor of the rule.

Heuristic rules also use functions which analyze cost estimates for individual representations and resulting canonical records to evaluate the efficiency of chosen designs. When the estimator is used to compute the cost of a skeleton (p. 41, SP 500-151), an itemized cost for accessing each entity and traversing each representation is computed. Heuristic rules may invoke functions in the their IF parts which do comparisons among their costs to identify relatively high and low cost records and relationship representations. Using this and other information, rules can infer the conclusions about the efficiency of individual records and representations.

Control Rules. The function of control rules is to conclude what actions are to be taken to process a problem, e.g. to control the operation of the KBS. These rules combine procedural knowledge and heuristics, relying on information about the current problem state to determine what action must be taken next. In the KBS, control rules are found exclusively in the Control Module (referred to as the High Level Control Module in SP 500-151). These rules may or may not use certainty factors. The following rule is an example taken from the Control Module Knowledge Base.

```
IF   (CLUSTER ?CLUSTER_ID ?CLUSTER_SET_ID)
     (CLUSTER_CHARACTERIZATION ?CLUSTER_ID ?CLUSTER_SET_ID EXTREMELY_LARGE)
     (COULD_NOT_CONCLUDE (REASON_NOT_TO_DIVIDE ?CLUSTER_ID ?CLUSTER_SET_ID) USER_DECLARED) )
     (COULD_NOT_CONCLUDE (REASON_NOT_TO_DIVIDE ?CLUSTER_ID ?CLUSTER_SET_ID) INTERSECTION_CLUSTER))
     (COULD_NOT_CONCLUDE (REASON_NOT_TO_DIVIDE ?CLUSTER_ID ?CLUSTER_SET_ID) AGGREGATE_IN_FORCE))

THEN (INITIAL_DECISION ?CLUSTER_ID ?CLUSTER_SET_ID DIVIDE_CLUSTER)
```

This rule states that if there is a cluster ?CLUSTER_ID characterized as being extremely large, and none of the list of conditions recommending against division exist, then the cluster should be divided.

2.4 Rule Groups And The Organization Of The Knowledge Bases.

As stated in the introduction, each knowledge base is subdivided into rule groups (Section 4.5 of SP 500-151). Rule groups are smaller collections of rules whose function is to make conclusions about information belonging to specific, defined subject areas. They may be thought of as having specific tasks or types of tasks to accomplish, such as the conclusion of certain types of facts which are used by other parts of the knowledge-based system. For example, all rules which conclude relationship representations on the basis of activity levels belong to one rule group. All rules which conclude relationship representations based on structural characterizations belong to another rule group. The relationship representations concluded by these rule groups are facts used to carry out design actions for record formation. Each rule group may contain one or more of the different rule types described in Section 2.2.

Knowledge bases contain rule groups whose subject areas and tasks are similar. For example, both of the rule groups mentioned above belong to the same knowledge base. Knowledge bases and rule groups are intended to provide an organizational framework useful for knowledge engineers and domain experts in understanding and maintaining the knowledge-based system. They are modular units which can be independently understood and maintained. As mentioned above, rule groups are often themselves subdivided into parts to make them easier to understand.

Individual rule groups can be thought of as being dependent on each other for information. In other words, one rule group may require facts, expressed in the antecedents of its rules, which are concluded by rules in another rule group. Rule group dependencies are extensive and cut across knowledge bases.

The two diagrams below show the knowledge bases and rule groups in the KBS, together with existing dependencies. Figure 1 shows all the knowledge bases and rule groups in the KBS, together with existing dependencies. Figure 1 represents a composite view of the entire system. It represents an updated version of the diagram which appears in Section 4.5 of SP 500-151. Since Figure 1 is complicated, selected parts of this diagram will be shown in subsequent chapters which cover individual knowledge bases.

In Figure 1 and in subsequent diagrams, dependencies between individual rule groups are depicted with thin black lines. If an individual rule group in one knowledge base is dependent on all the rule groups in another knowledge base, a thick light line is drawn between the rule group and the knowledge base. Similarly, if all the rule groups in one knowledge base are dependent on all the rule groups in another knowledge base, a thick light line is drawn between the knowledge bases. An arrow points from the dependent rule group or knowledge base to the rule group or knowledge which it is dependent on. Bidirectional dependencies are also possible.

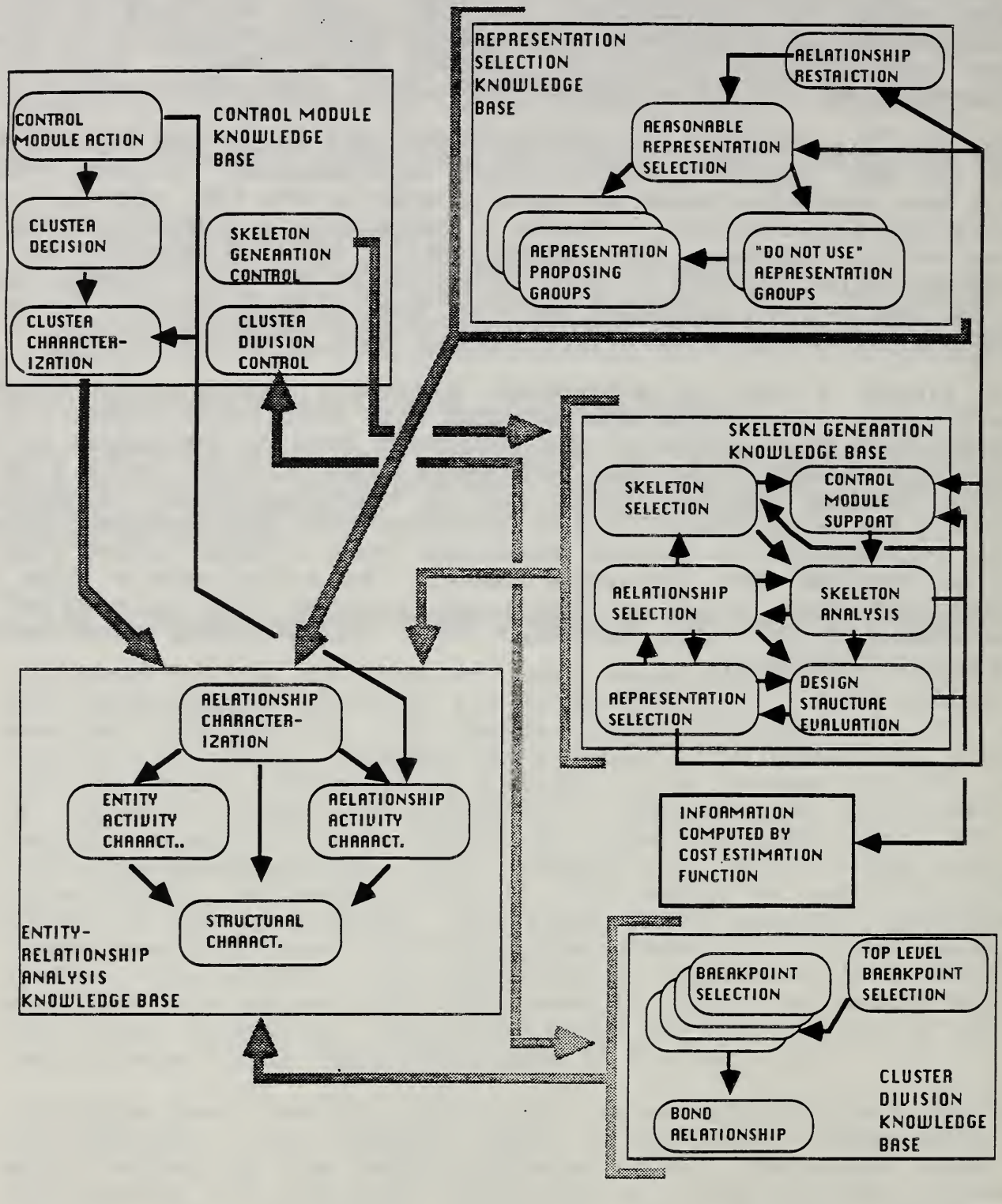


Figure 1. Diagram Of Dependencies In The KBS

Some rule groups rely on given facts about the LDS, the workload, or on facts computed by algorithmic routines. This is not shown in Figure 1. Other rule groups are dependent entirely on such information to make inferences and do not depend on other rule groups. For instance, the Structural Characterization Rule Group in the Entity Relationship Analysis Knowledge Base relies only on the problem description for information.

Invocation of a rule group may result in consideration of all rules in the rule group, or in consideration of a subset. A particular rule group may be invoked only once during the processing of a problem, as in the case of the rule groups in the Entity Relationship Analysis Knowledge Base. Or, a rule group may be invoked repeatedly to redetermine facts, as in the case of the rule groups in the Control Module Knowledge Base.

The rule groups described in this report are almost the same as those outlined in SP 500-151. However, the KBS has evolved in the time since the previous publication was issued. Changes and additions have been made.

3. THE CONTROL MODULE KNOWLEDGE BASE

The Control Module Knowledge Base[1] consists of five rule groups. The Control Module Action Rule Group is the highest level rule group in the KBS. Its responsibility is to determine which global design action to take next on the problem being processed, e.g. which action on what cluster or what part of the problem. The Cluster Decision Rule Group is invoked as a result of a decision by the Action Rule Group to determine an action on an individual cluster being processed. The Cluster Characterization Rule Group is used by both of the other rule groups to reach decisions. The purpose of this rule group is to provide specific characterizations of individual clusters. See Section 4.5.1. of SP 500-151.

The last two rule groups pertain to control of cluster division and skeleton generation. The Cluster Division Control Rule Group (Section 5) is a small set of rules for invoking individual breakpoint selection rule groups (described in Section 5 of this report and Section 4.5.4 of SP 500-151). The Skeleton Generation Control Rule Group (Section 6) determines actions for selectively generating skeletons. Discussion of these two rule groups will be postponed until Section 5 and Section 6 where the context will be clearer.

The rules for this knowledge base are found in Appendix A. Figure 2 below shows the dependencies for the rule groups in the Control Module Knowledge Base.

---

[1]In SP 500-151, the Control Module Knowledge Base was called the High Level Knowledge Base. The Control Module was referred to as the High Level Control Module, or the High Level.

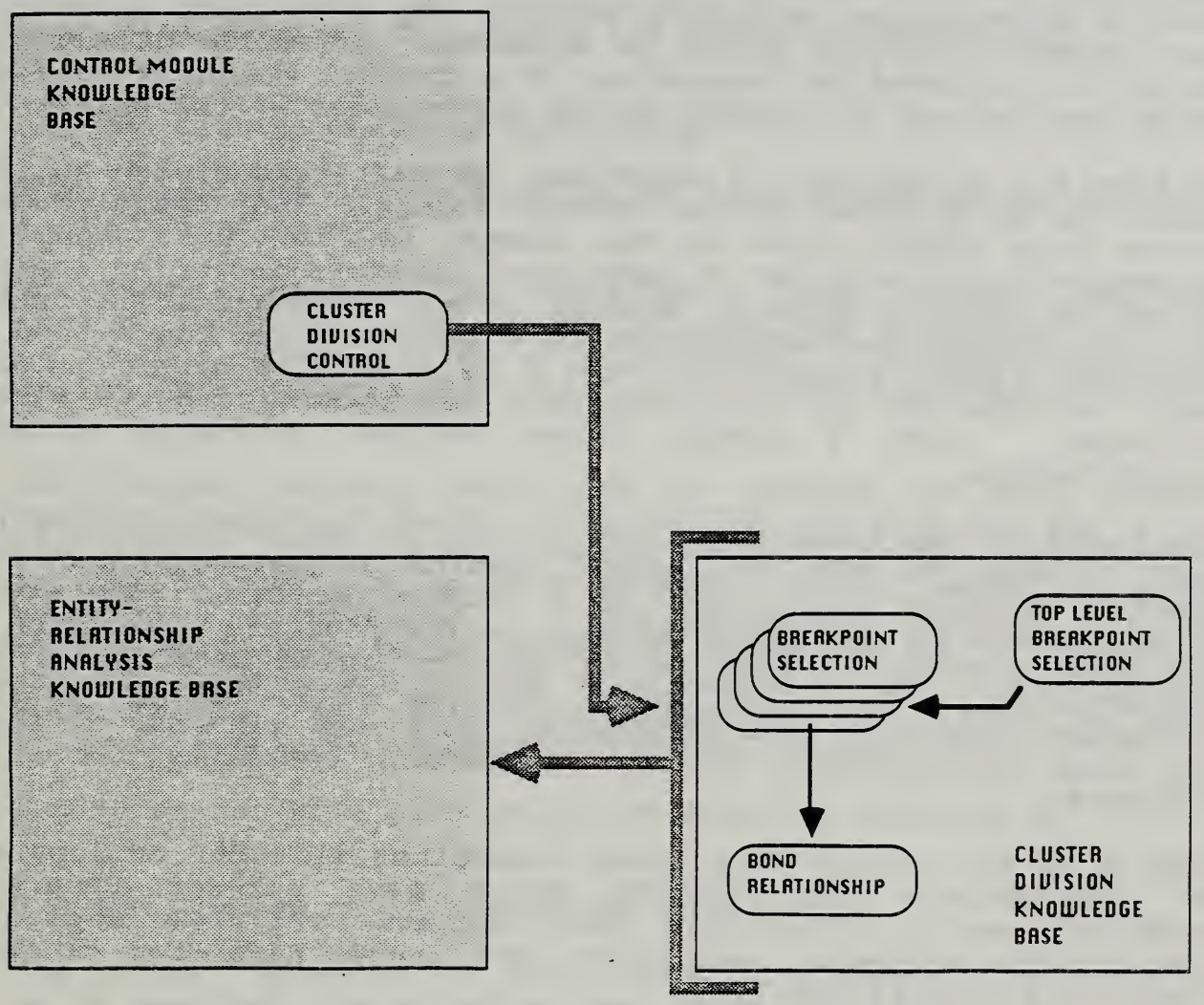


Figure 2. Dependencies Between Rule Groups In The Control Module Knowledge Base

3.1 The Control Module Action Rule Group

The Control Module Action Rule Group concludes design actions. The rules in this rule group recommend individual design actions with an accompanying certainty factor. The rule which fires with the highest certainty factor is selected. Selection of a design action is followed by invocation of a forward chaining rule set which carries out the design action recommended by the rule.

Section 4.1 of SP 500-151 describes the different design actions which may be selected by the Control Module. The Control Module Action Rule Group relies on the Cluster Decision Rule Group for determination of actions on individual clusters including whether or not to divide a cluster, whether or not to restrict the number of relationship representations in a cluster, and the method of canonical record formation. Part A of this rule group concerns basic actions for dividing clusters and forming records within clusters. Part B covers other actions including cluster recombination.

The tasks of the rules described in Part A include selecting the next cluster to work on, which results in invocation of the Cluster Decision Rule Group by a forward chaining rule for a determination of the action to take on the cluster. Selection of the next cluster to work on is done by rules CM_ACTION_2 and CM_ACTION_3. These two rules may be triggered by several different clusters, each requesting that work be done on them. The rule instantiation having the highest certainty factor is selected.

Part B of the Action Rule Group determines if other actions may be appropriate for clusters which have already undergone canonical record formation. This includes revisiting a cluster for further skeleton generation and recombining two clusters (Section 4.4 of SP 500-151).

During problem processing, skeleton generation for a particular cluster may result in creation of fewer skeletons than the maximum number specified by the rules in Part C of the Cluster Characterization Rule Group (Section 3.3 of this report). In this case, there will be spare skeletons which can be reassigned to another cluster.

In general, overall problem processing proceeds as follows. An LDS is broken down into clusters of reasonably small size. Each cluster has its most efficient skeletons and canonical records determined. If there are spare skeletons, certain clusters may be revisited for more skeleton generation. Recombination rules then begin firing for adjacent clusters, resulting in formation of temporary intersection clusters. The temporary clusters then receive record formation actions. This continues until the

certainty of rules recommending different recombinations is less than the certainty of rules CM_ACTION_10 and CM_ACTION_11 recommending that fine-tuning should be invoked. At this point KBS processing ends.

The rules for this rule group are found in Appendix A.

3.2 The Cluster Decision Rule Group

This rule group is responsible for determining design actions for individual clusters.

There are two types of decisions. Each cluster receives an initial decision. The rules for this are described in Part A. If further processing is still required, a follow-up decision is called for by the Action Rule Group. This is described in Part B.

The Cluster Decision Rule Group is invoked for each cluster to make an initial decision. The initial decision may be to divide the cluster, restrict the number of relationship representations in the cluster, or to form records through selective skeleton generation enumeration. The follow-up decision may be to selectively generate or enumerate skeletons for a cluster which did not undergo this action as a result of the initial decision. Both initial and follow-up decisions are based on cluster characterizations, discussed in detail in Section 3.3.

The rules for this rule group are found in APPENDIX A.

3.3 The Cluster Characterization Rule Group

The primary function of the Cluster Characterization Rule Group is to conclude characterizations about individual clusters. The conclusions of this rule group are used by the Action Rule Group and the Cluster Decision Rule Group.

Part A contains rules for characterizing clusters on the basis of size and the internal workload complexity between interrelated entities in the cluster (Section 4.2 of this report).

Part B consists of several rules to determine the relative priorities of clusters. These priorities are determined by the relative size and the amount of workload complexity for individual relationships within the cluster. The priority level is reflected in the assigned certainty factor of the rule. This information is used by the Control Module Action Rule Group to determine the order in which to work on clusters.

Part C has four rules to determine the number of skeletons to be

generated within a cluster. This determination is separate from the number of skeletons to be generated during selective skeleton generation (a different determination described in Section 7.3 of this report). This number will be based on the total number of skeletons to be generated in the entire problem (provided by the user at the start of a session), the size of the cluster in terms of the number of relationships, and the existence or absence of workload complexity within the cluster. That is, a cluster with workload complexity should have more skeletons generated than one without workload problems.

The rules for this rule group appear in APPENDIX A.

4. DESCRIPTION OF THE ENTITY RELATIONSHIP ANALYSIS KNOWLEDGE BASE

The purpose of this knowledge base is to provide heuristics for analysis and characterization of LDS structure, activity levels and workload complexity. The rule groups in this knowledge base are invoked once at the beginning of problem processing.

The rules for this knowledge base are found in Appendix B. Figure 3 shows dependencies for the rule groups in this knowledge base.

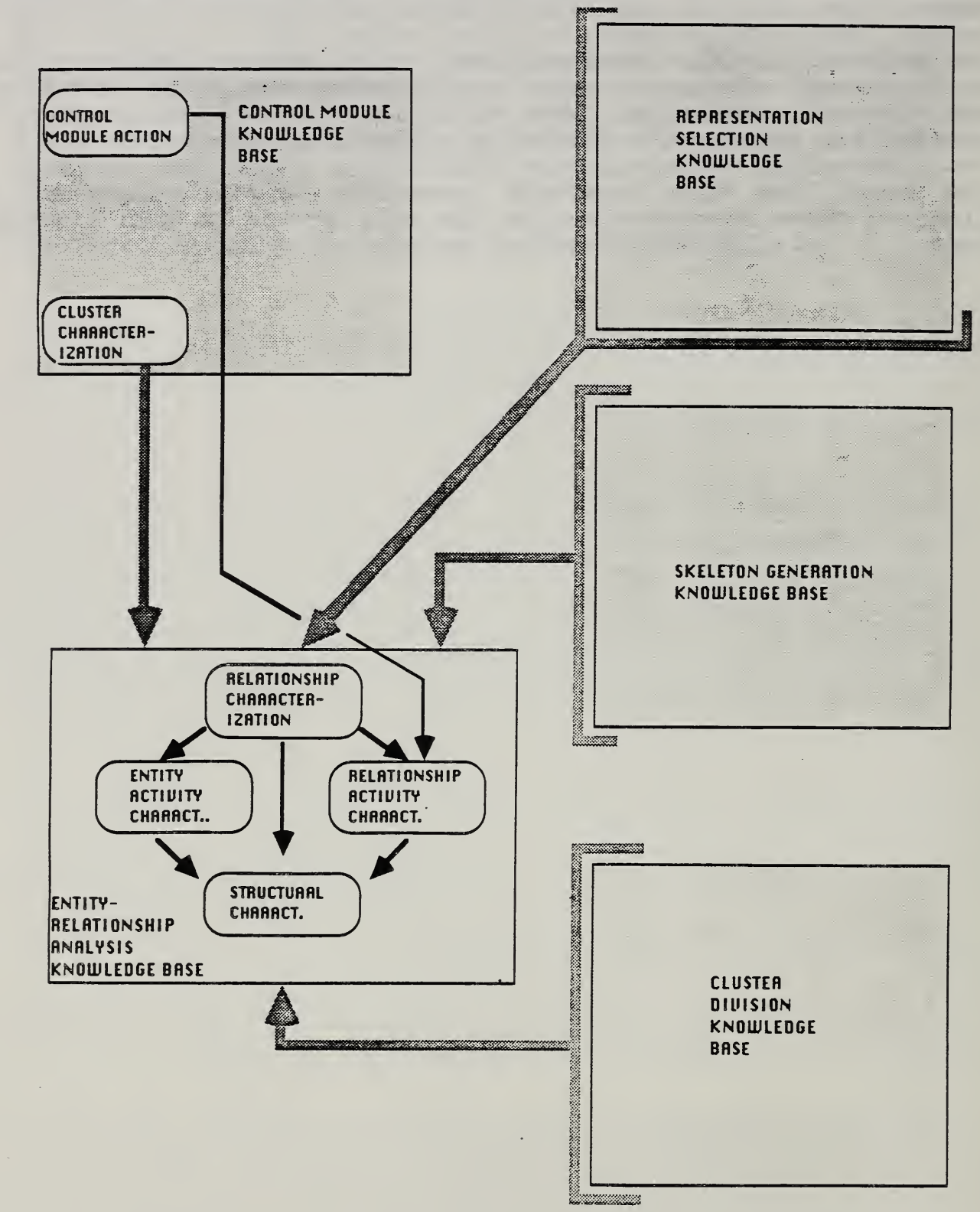
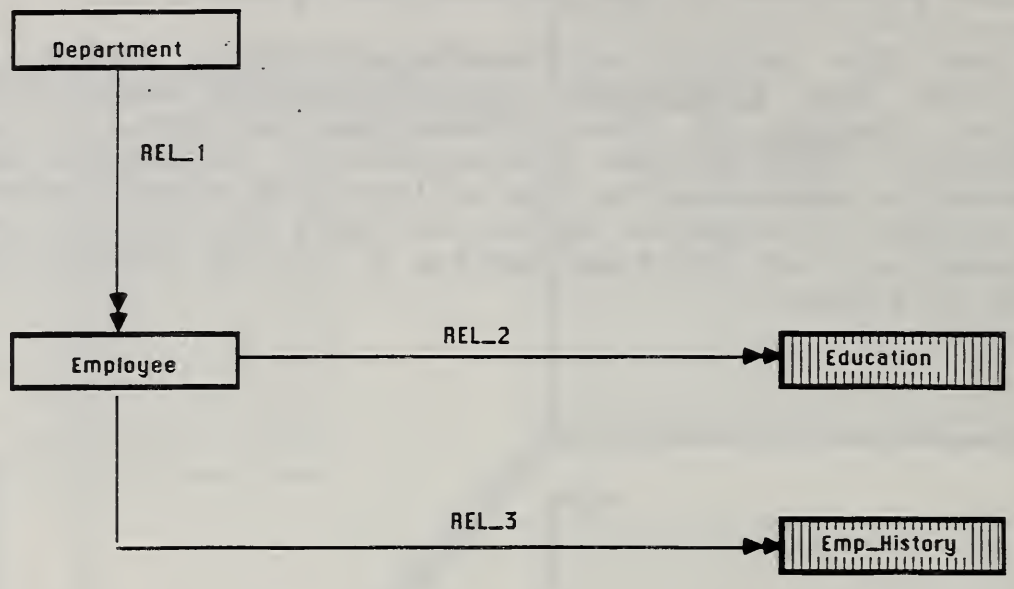


Figure 3. Dependencies Between Rule Groups In The Entity Relationship Knowledge Base

4.1 The Structure Of The Information System.

We use the entity-relationship-attribute model [CHEN76] to represent the structural characteristics associated with the logical design of the information system. A Logical Data Structure (LDS) is the description of a logical design of an information system using the entity-relationship-attribute model. For further discussion on Logical Data Structures, see [CHEN76]. An example of an LDS from Section 2.1 of SP 500-151 is reproduced below in Figure 4.



This diagram displays three relationships which connect four entities. One to many relationships are represented by double arrows drawn near the "M" entity. For example, there may be many instances of EMPLOYEEs for an instance of a DEPARTMENT, but there will be only one instance of a DEPARTMENT for any given instance of EMPLOYEE. A relationship may be described in two directions with each direction having a unique descriptor name derived from the primary identifiers of the entities. For example, the descriptor names for REL_1 are EMPLOYEES_OF_DEPARTMENT and DEPARTMENT_OF_EMPLOYEES.

Entities in boxes with vertical stripes are dependent entities. For the purposes of this example, a dependent entity will be a "many" entity which is dependent on a "1" entity in a one to many relationship (i.e., the mapping is onto). An entity that is not dependent on any other entity is an independent entity. Independent entities are in unstriped boxes. DEPARTMENT and EMPLOYEE are independent entities, while EDUCATION and EMP_HISTORY are dependent entities. See Section 4.4 for more complete definitions of independent and dependent entities.

Figure 4. A Small Portion Of An LDS Diagram

4.2 The Workload Of The Information System.

Workload refers to the retrieval and update activity associated with an information system. It is a critical and very complicated variable in the physical design process. Characterizations about activity levels and workload complexity, briefly discussed in Section 2.2 of this report, are obtained from analysis of workload. These characterizations are used by the KBS for physical design in selecting physical level schema. Section 2.1.2 of SP 500-151 discusses workload. That discussion will be repeated and expanded upon in this report.

Figure 5 represents the retrieval workload for the small LDS portion shown in Figure 4. Four retrievals are shown. Database Language SQL [ANSI86], plus quantitative parameters, has been used in this example, because SQL is a commonly used, standard database language[2]. A complex retrieval is composed of a number of contexts, each of which deals with one entity (but possibly retrieving many instances of that entity). Individual contexts are described by selection criteria, projection criteria, and ordering criteria. Further discussion of context may be found in [CARL80].

Each context has an associated frequency. Frequency refers to the number of times the context is executed per month. Each context also specifies the average proportion of record instances retrieved during each execution. Retrieval activity is forwarded from one context to the next; that is, the activity continues at the latter entity in the context of the former. For example, {Retrieval 1} of Figure 5 has two contexts; the first dealing with EMPLOYEE and the second with EDUCATION. The EMPLOYEE context is executed 500 times per month with 25% of the instances retrieved during each execution. Activity is forwarded to the EDUCATION context which is executed 12500 times per month with 25% of the instances retrieved during each execution. Context may be very important for performance. In {Retrieval 1}, the retrieval of a large number of EDUCATION records in the context of a specific EMPLOYEE can be much less costly than the retrieval of a similar number of random EDUCATION records.

---

[2]The actual language used in our system is navigational [CARL80]. A future enhancement would be a query optimization phase that would translate SQL into a navigational form. The reader can assume that navigation follows the order in which entities are listed in the SQL. For example, {Retrieval 1} starts at EMPLOYEE and navigates to EDUCATION.

```
{Retrieval 1}
SELECT    EMPLOYEE_NAME, SSN, INSTITUTION_NAME, MAJOR
     FROM      EMPLOYEE, -- {FREQUENCY 500, PROPORTION 0.25}
          EDUCATION -- {FREQUENCY 125000, PROPORTION 0.25}
     WHERE     EDUCATION.EMPID = SSN AND AGE < 30


{Retrieval 2}
SELECT    *
     FROM      EMPLOYEE  -- {FREQUENCY 1000, PROPORTION 1.0}


{Retrieval 3}
SELECT    DEPARTMENT_NAME, EMPLOYEE_NAME, SSN, AGE
     FROM DEPARTMENT,    -- {FREQUENCY 500, PROPORTION 0.5}
          EMPLOYEE  {FREQUENCY 300000, PROPORTION 0.3}
WHERE     EMPLOYEE.DEPT = DEPARTMENT_NAME AND AGE >50


{Retrieval 4}
SELECT    EMPLOYEE_NAME, SSN, EMPLOYER, START_DATE, END_DATE
     FROM      EMPLOYEE  -- {FREQUENCY 10, PROPORTION 0.001}
          EMP_HISTORY {FREQUENCY 10, PROPORTION 0.001 }
   WHERE EMPLOYEE.SSN = ?
```

All retrievals have multiple contexts except {Retrieval 2}. {Retrieval 4} contains a small subset retrieval of EMPLOYEE. {Retrieval 1} is a large subset retrieval with two contexts. The first is a retrieval on EMPLOYEE, selecting individuals under 30 years of age. This context is executed 500 times per month, and 0.25 of the EMPLOYEE records are retrieved in each execution. The second context is on EDUCATION, retrieving those instances belonging to qualifying EMPLOYEE instances. This context is retrieved with a frequency of 125000 times per month, with 0.25 of the EDUCATION records retrieved in each execution.

Figure 5. Retrievals for Figure 4.

Retrieval size is a classification of contexts on the basis of proportion of instances retrieved. A single record retrieval means only one record in every execution of the context. A small subset retrieval is the retrieval of a small proportion of a large file which requires a secondary index for efficient search (usually less than 10% of a file). A large subset retrieval is the retrieval of a sufficiently large proportion of records of a large file (over 10%) to require a scan of the entire file for efficient search. In the KBS, retrieval context frequencies which focus on entities and traverse relationships are totaled according to retrieval size.

Update workload is also important. It is described by the frequency of insertion and deletion of instances of entities and the frequency of modification of entity attributes. Updates are assumed to be singular, that is, there are no small subset or large subset updates. Frequencies for updates are also totaled.

There are two important aspects to workload analysis: 1) determining the relative frequencies of different retrieval and update activities to identify entities and relationships having relatively high and low activity levels, and 2) determining activity levels along relationships on the basis of the proportion or percentage of activity forwarded from one entity to the other. The first is called absolute activity level and the second is known as forwarding percentage.

Figure 6 which is based on the retrievals in Figure 5 below shows the total forwarding percentage in the sample.

Retrieval Number	Initial Entity & Frequency	Subset Size	Subsequent Entity & Forwarding Relationship
1	EMPLOYEE (500)	Large	EDUCATION (REL_2)
2	EMPLOYEE (1000)	Large	NONE
3	DEPARTMENT (500)	Large	EMPLOYEE (REL_1)
4	EMPLOYEE (10)	Single	EMP_HISTORY (REL_3)

Forwarding Percentage For Relationships
(From the one entity to the many entity only)

Name	Single	Small Subset	Large Subset
REL_1	n/a	n/a	1.0
REL_2	0.0	n/a	0.333
REL_3	1.0	n/a	0.0

* n/a means no activity of this subset size was forwarded along this relationship.

Figure 6. Summary Of Absolute Activity Levels And Forwarding Percentage For Retrievals In Figure 5.

Workload analysis is, in part, the process of identifying areas of high retrieval and update activity or lack thereof, and making generalizations and characterizations about this activity. The result of this analysis is a set of absolute activity characterizations for the entities in the LDS and a set of absolute activity and forwarding percentage characterizations for the relationships in the LDS. The Entity Activity Characterization Rule Group and the Relationship Activity Rule Group perform these functions.

Workload analysis also indicates the existence of workload complexity. Workload complexity (Section 2.1.2 of SP 500-151) is a measure of the number of different retrievals, the variability of their size and frequency, and the amount of update workload. The importance and complexity of physical database design increases rapidly as workload complexity increases. The Relationship Characterization Rule Group identifies individual relationships and groups of relationships having more extensive workload complexity.

4.3 The Organization Of The Entity Relationship Analysis Knowledge Base.

Definition of structural features and structural characterizations is performed by the Structural Characterization Rule Group, described in Section 4.4. The Entity Activity Rule Group, covered in Section 4.5, characterizes direct retrieval activity, or initial context activity, which focuses on individual entities. The Relationship Activity Rule Group, Section 4.6, characterizes absolute activity levels and forwarding percentage for individual relationships. Characterization of workload complexity problems resulting from combinations of structural and activity characterizations is the function of the Relationship Characterization Rule Group, described in Section 4.7.

The Entity Activity Characterization Rule Group and the Relationship Activity Characterization Rule Group are dependent on the Structural Relationship Activity Rule Group. The Relationship Characterization Rule Group is dependent on each of the three previous rule groups. The information concluded by these rule groups is used by many other rule groups in different knowledge bases.

4.4 The Structural Characterization Rule Group

This rule group contains definition rules for structural features such as relationship degree, entity type, and entity dependency. These rules use data on the entities, relationships, attributes, and entity identifiers, and use this information in the IF parts

of the rules. Structural characterizations based on these structural features are also concluded by several characterization rules.

Part A of this rule group includes rules for concluding partially identifying relationships and the existence of dependency between entities.

Part B is concerned with entity type and characterizations based on entity type. In [CARL80], there are four recognized entity types: independent, aggregate, dependent, and intersection. Independent entities have primary identifiers which are composed entirely of their own attributes, and do not contain the attributes of any other entity. Thus, independent entities do not have partially identifying relationships. Furthermore, an independent entity must have at least one attribute which is not part of its identifier. Rule STRUCT_6 in Appendix B defines an independent entity. Aggregate entities, Rule STRUCT_11, have primary identifiers which are composed of all of their attributes and do not have partially identifying relationships. In contrast to independent entities, aggregate entities have no attributes which are not part of their identifiers. Dependent entities, Rule STRUCT_13 in Appendix B, have identifiers composed of at least one (but not all) of their attributes and have one and only one partially identifying relationship. They are dependent on the other entity in the partially identifying relationship (represented with a DEPENDENT_ON fact expression). Intersection entities, Rule STRUCT_14, have identifiers composed of two or more partially identifying relationships (also represented by DEPENDENT_ON fact expressions). The function of intersection entities is to map many to many relationships between two other entities.

Part C is concerned with conclusions about relationship degree, already partially introduced in Section 2.2. Two kinds of conclusions about relationship degree may be made. Either a relationship between two entities is one to one, that is, there is one instance of each entity associated with each instance of the other. Or, the relationship is one to many, meaning that one of the two entities has many instances associated with a single instance of the other. Relationships where the degree is zero or one to one are considered to have one to one degree. Many to many relationships are not explicitly represented. Instead many to many relationships are represented by intersection entities, defined above.

The rules in Part D make determinations about small and large record sizes which would occur if absorption should take place along certain relationships. This information is used by the Relationship Representation Knowledge Base for recommending reasonable representations when device length restrictions are in effect.

Part E has rules for miscellaneous conclusions not covered in the previous three parts. These rules are concerned with identifying attributes of dependent entities which are transferred to these entities along partially identifying relationships. This information is used by algorithmic routines for computation of record lengths invoked by the Cost Estimation Function (Section 4.5.6 of SP 500-151). Part E also has rules for calculating primary segment and secondary segment items for physical records based on the "eighty twenty" rule. See [TEOR82] for a detailed discussion of segmentation and the "eighty twenty" rule.

The rules for this rule group are found in Appendix B.

4.5 The Entity Activity Rule Group

The function of this rule group is to conclude characterizations about workload which focuses directly on entities, e.g. initial context retrievals.

The rules in Part A conclude characterizations about activity levels for direct single, direct small subset, and direct large subset retrievals. This is done by comparing the frequency of direct activity on an entity with the total frequency of retrieval forwarded to the entity along all the relationships the entity is in. The frequencies are adjusted on the basis of subset size. If the adjusted frequency of direct activity is equal to, or greater than, a constant proportion of the adjusted frequency of the activity forwarded along the relationships, or is within a specified range, an activity level characterization is concluded.

Part B contains rules for numerical computation of total activity frequencies. These are numeric computation rules which invoke functions to perform the actual computation of direct single, small subset, and large subset activity on individual entities.

The rules for this rule group are found in Appendix B.

4.6 The Relationship Activity Characterization Rule Group

The function of this rule group is to conclude characterizations about absolute activity levels which focus on entities along relationships, e.g. subsequent context retrievals.

Part A of this rule group contains rules which characterize absolute activity levels for retrieval and update activity in increasing order of magnitude as SIGNIFICANT, and HEAVY-0, HEAVY-1, HEAVY-2, HEAVY-3, or HEAVY-4.

Part B contains rules for concluding characterizations about forwarding percentage. The forwarding percentage (Section 4.2 of this report) may be characterized in increasing order of magnitude as SIGNIFICANT-FORWARDING, HEAVY-FORWARDING-0, HEAVY-FORWARDING-1, HEAVY-FORWARDING-2, HEAVY-FORWARDING-3, or HEAVY-FORWARDING-4.

The rules in Part C determine computed frequency totals for individual relationships for forwarding and nonforwarding retrieval and for forwarding percentage. These are numeric computation rules, which make function calls to do the summation. Individual computations are made for single, small subset, and large subset retrieval.

Part C also contains rules which determine minimum threshold levels for activity forwarded along a relationship, below which the activity cannot be characterized as significant. These determinations appear in the rules in Part A.

The rules for this rule group are found in Appendix B.

4.7 The Relationship Characterization Rule Group

The function of this rule group is to identify relationships with significant workload complexity which require special consideration when choosing reasonable representations or selecting breakpoints. Characterizations about activity levels focusing on entities directly or forwarded to entities along relationships, characterizations about forwarding percentage, and structural characterizations are used by this rule group. This rule group is dependent upon the other three rule groups in this knowledge base.

Part A is concerned with identifying special situations which arise from high activity levels. This includes relationships with significant activity in both directions, significant activity in the many to one direction only (a situation which suggests the use of many to one direct pointers as recommended in the Complex Representation Proposal Rule Group, Section 5.8), the existence of significant or heavy nonforwarding activity in the one to many direction (a special case handled by the same rule group), and the existence of significant relationship update activity.

Part B is concerned with identifying relationships in which one of the entities is subject to context conflict. Context conflict occurs when an entity is subject to large subset retrieval via two or more paths, each of which requires a different ordering of the file.

The rules for this rule group are found in Appendix B.

5. DESCRIPTION OF THE REPRESENTATION SELECTION KNOWLEDGE BASE.

The objective of representation selection is the identification of relationship representations which will lead to formation of efficient canonical records.

In this section, a discussion of relationship representations will first be presented. This topic was covered in Section 2.2 of SP 500-151, but will be restated and expanded upon. The function of the representation selection rules, their limitations, their role in the KBS, and the general strategy of these rules will then be presented. Finally, the seven rule groups of this knowledge base will be described.

The rules for this knowledge base are found in Appendix C. Figure 7 shows a diagram of dependencies for the rule groups in this knowledge base.

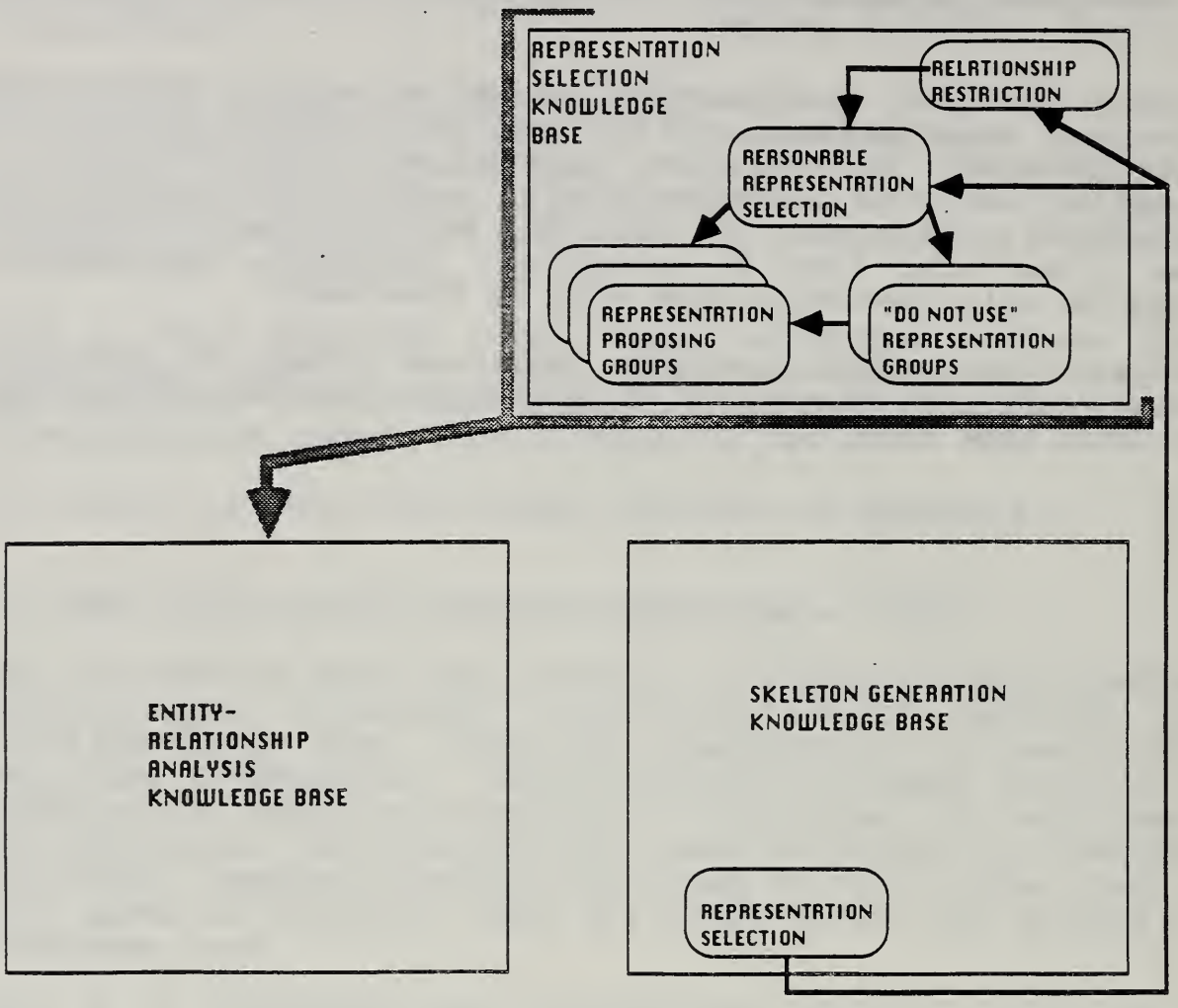


Figure 7. Dependencies Between Rule Groups In The Representation Selection Knowledge Base

5.1 The Combinatorics of Selection of Representations

Selection of representations provides the basis for the formation of physical records and for the physical relationships among those records. Representations are of three generic types:

- o absorption, where the two entities of a relationship are stored in the same physical area;
- o symbolic pointer, where one entity contains the logical identifier of the other; and
- o direct pointer, where one entity contains the physical address of the other.

These representations may be used in combination; e.g., both a direct and a symbolic pointer could be used to represent the relationship from a dependent entity to an entity on which it is dependent. Carlis [CARL80] has identified 10 possible combinations of the generic representations for relationships involving dependent entities and 17 possible combinations for relationships where no dependency between entities exists.

In practice, more than one representation is usually selected for each relationship. In a large LDS, the number of potential combinations which results is enormous. Let us say, for example, an average of three representations is chosen for each relationship in an LDS which contains 100 relationships. The number of possible combinations of relationship representations becomes 3**100. More discussion of this issue can be found in Sections 2.2 and 2.3 of SP 500-151 and in [CARL80].

Each combination is referred to as a skeleton. Each skeleton contains a unique combination of relationship representations, e.g. physical records for the cluster. Figure 8 shows two skeletons with their records. The skeletons differ by the representation of relationship REL_3. In Skeleton 1, relationship REL_3 is represented by absorption. In Skeleton 2, the representation is a symbolic pointer. As a result, Skeleton 1 has two canonical records (marked by boxes with shaded lines), while skeleton 2 has three canonical records.

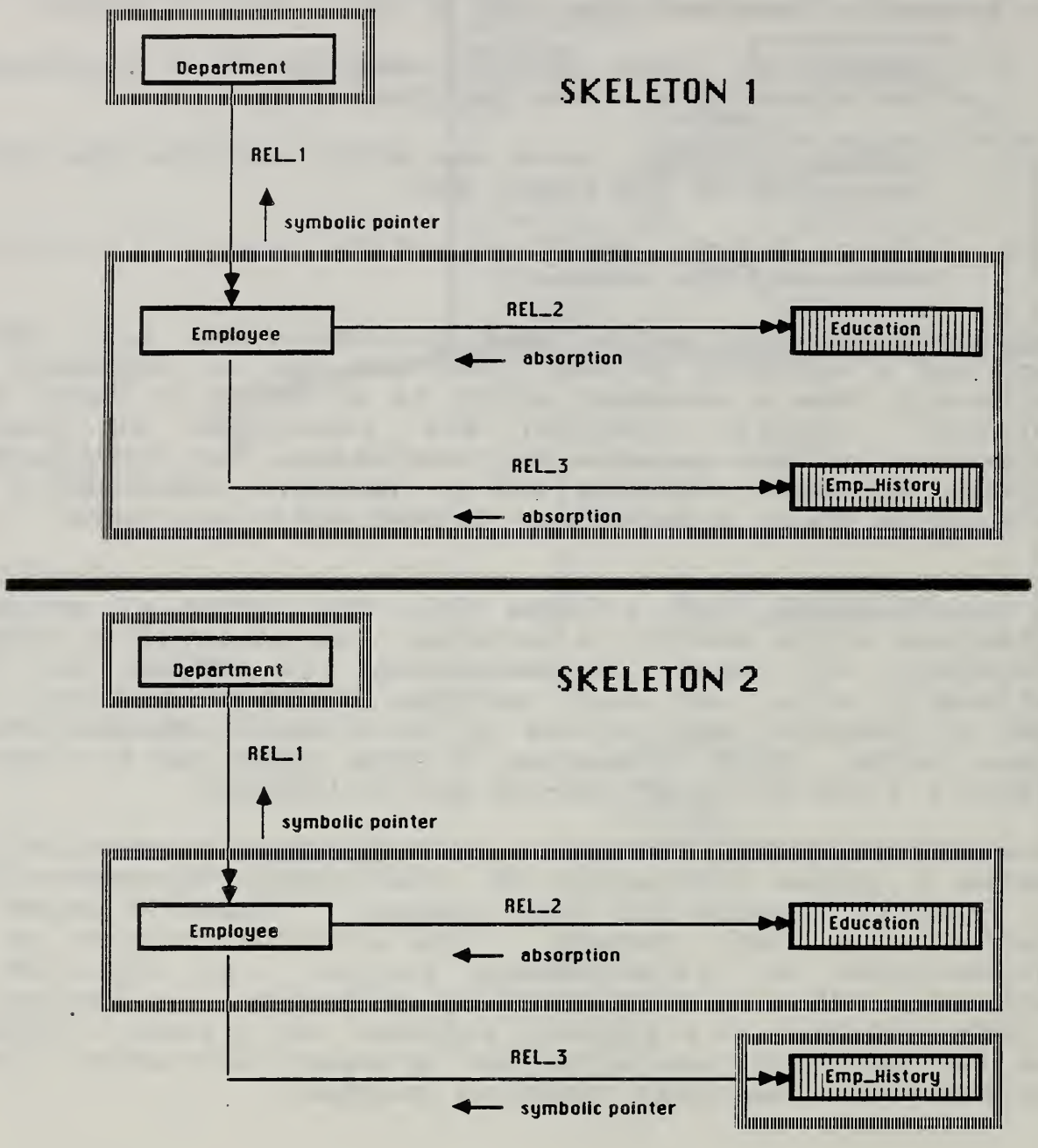


Figure 8. Two skeletons For The LDS Diagram

Each skeleton has a different cost for processing the retrieval and update workload, with some skeletons having very efficient canonical records and others having records with prohibitive costs. Since the identity of individual canonical records is determined by relationship representations, the selection of these representations is critical.

5.2 Representation Selection Rules

The task of the rules for representation selection is to identify the representations likely to be the most efficient when skeletons and canonical records are formed.

Rules for selection of representations, as well as for selection of breakpoints, have limitations. The representational capacity of a rule as a data structure is limited. No more than a few clauses about relationships and entities can be explicitly represented in one rule, otherwise rules would be too large to read and understand. In practice, therefore, the rules in the KBS usually address no more than 2-3 relationships and 2-3 entities.

Often, there can exist many feasible combinations of activity characterizations and workload complexity characterizations within a group of entities in relationship to each other. In such cases, the number of rules required to cover all the possible combinations of variations in activity patterns and complexity characterizations could yield a very large and possibly unmanageable rule set. This is especially so if the rules are meant to address a larger group of entities in relation to each other.

However, in practice it is known that workload complexity may extend to several entities along several relationships. To cope with this, a large number of rules with many clauses would probably be necessary, the development of which is beyond the scope of current technology. The rules in the KBS are more limited in scope.

5.3 The Organization Of The Representation Selection Rule Groups

The rule groups of this knowledge base implement the strategy discussed above. The rule groups have five functions: 1) proposing representations with an associated certainty factor, 2) asserting certain representations should not be used with an associated certainty factor, 3) asserting representations should not be used under any circumstances, e.g. absolute prohibition, 4) determining an initial set of reasonable representations and associated certainty factors for each relationship in the LDS, to be considered during further problem processing, and 5)

implementing direct reduction of problem size for individual clusters, e.g. relationship restriction.

5.4 The Structural Characterization Proposal Rule Group

The function of this rule group is to propose relationship representations on the basis of structural characterizations and storage efficiency considerations, with very little reliance on information about workload. As such, this rule group is heavily dependent on conclusions provided by the Structural Characterization Rule Group (Section 4.4 of this report). The group consists of some 19 heuristic rules. The three generic representations (symbolic pointers, direct pointers, and absorption) are recommended by this rule group.

This rule group, as well as the Structural Characterization Do Not Use Rule Group presented in the next section, is based largely on the system of heuristics developed by John Carlis in [CARL80].

The conclusions of the rule group are used by the Reasonable Representation Rule Group, described in Section 5.9. The application of this rule group is by itself not adequate to produce a sufficiently small number of good relationship representations which would result in a small number of alternative skeletons to examine. After this rule group has been applied to even a moderately sized problem, there are still far too many alternative skeletons to examine. See [AUER81] for an example of the results of using structural heuristics alone.

Rule groups for selection of representations based on workload factors and for preventing selection of relationship representations help reduce problem size by assigning high or low certainty factors to individual relationship representations. This permits further pruning of certain alternatives when design actions for canonical record formation are invoked. These rule groups are necessary to adequately limit the search space.

The rules for this rule group are found in Appendix C.

5.5 The Structural Characterization Do Not Use Rule Group

The function of this rule group complements that of the Structural Characterizations Proposal Rule Group. Its purpose is making recommendations that particular representations should not be used on the basis of structural factors and to minimize redundancy. The scope of this rule group is limited to the three generic representations.

One may notice that several rules are concerned with selection of

representations in circumstances which are not covered by rules which propose representations. For instance, no rule proposes absorption of a "one" entity into a "many" entity, yet there is a rule which prevents this selection. Rules such as these may therefore appear not to have any practical purpose. However, it is still useful to include such rules for the sake of completeness in the knowledge base, and as a matter of record. Also, the rules would be useful if a user manually added representations to the KBS during a problem solving session.

Another heuristic used by this rule group involves forming records which are larger than the storage capacity of a unit of a physical device, such as a track on a DASD device. Long records often cause problems in such devices, resulting in delays. Each of the heuristics described above has been translated into one or more specific rules.

The rules for this rule group are found in Appendix C.

5.6 The Activity Characterization Proposal Rule Group

This rule group proposes relationship representations primarily on the basis of workload characterizations. Structural characterizations play a supporting role.

Most of the rules are concerned with recommending absorption, with one rule recommending symbolic pointers and one rule for direct pointers. The primary function of these rules is to recommend absorption for relationships with high activity and high forwarding percentage. The certainty factors of these rules are combined with the certainty factors of the Structural Proposal Rule Group according to the Bernoulli formula (and may be offset by determinations by any Do Not Use rules) to strengthen the overall certainty of absorption for relationships with high activity characteristics. See Section 5.9 of this report or Section 3.1 of SP 500-151 for further discussion of certainty factor combination.

Representations having high certainty are tried first during skeleton generation (Section 6 of this report), thus improving the search process. Representations with lower certainties are tried later or not at all. If relationship restriction is applied to a cluster, some representations for relationships with low certainty factors may be eliminated to reduce the size of the search space (Reasonable Representation Rule Group, Section 5.9 of this report).

The rules for this rule group are found in Appendix C.

5.7 The Activity Characterization Do Not Use Rule Group

This rule group consists of heuristic rules which provide recommendations that a relationship representation should not be used. The rule group is mainly concerned with identifying situations in which activity characterizations indicate that absorption is not a good representation.

The cumulative effect of these rules is to produce a recommendation against absorption which has a certainty related to the levels of direct large subset activity and forwarding percentage. The function *WORK-STRENGTH* uses workload levels to determine the certainty factor within a specified range.

If device length restrictions are in effect, the effect of large file sizes which would be produced by absorption also results in recommendations against absorption with an associated high certainty factor.

The rules for this rule group are found in Appendix C.

5.8 The Complex Representation Proposal Rule Group

This rule group consists of heuristic rules which recommend special, seldom used, representations as well as the three generic representations to handle relationships with significant workload complexity characterizations or with nonforwarding context activity.

Part A of this rule group covers representations for relationships with characterizations of certain workload complexity problems identified by the Relationship Characterization Rule Group (Section 4.7 of this report).

Part B of this rule group deals with relationships having significant nonforwarding context activity. The Relationship Characterization Rule Group and the Relationship Activity Rule Group (Section 4.6 of this report) makes characterizations pertaining to this type of activity. The representations recommended include one to many symbolic pointers, and symbolic pointers in both directions of a relationship.

The rules for this rule group are found in Appendix C.

5.9 The Reasonable Representation Rule Group

This rule group combines the conclusions of the previous rule groups to arrive at a determinations about whether particular relationship representation can be used in further processing for a design problem, with a specific target database management

system in mind. The objective of this rule group is twofold: 1) to determine which relationship representations are reasonable and consistent with the target database management system, allowing them to be used in subsequent processing, 2) to translate individual relationship representations into the particular format of the target database management system.

Part A selects reasonable representations which are consistent with the target database management system. For a particular representation to be selected, it must be recommended by at least one rule in one of the representation proposal rule groups (Sections 5.4, 5.6, and 5.8) and must not have an ABSOLUTE_PROHIBITS fact expression concluded (Rules PREV_S_9, PREV_S_10, and PREV_S_11 of Section 5.5 of this report). If no proposal rules fire, Rule REAS_5 provides a default representation. The proposal rules provide a positive certainty factor for the use of a rule. Do Not Use recommendations (Sections 5.5 and 5.7), if they exist, have the effect of lowering the certainty factor associated with the representation. The Bernoulli Formula is used to combine the certainty factors. See Section 2.2 of this report for an explanation of this formula. Section 3.1 of SP 500-151 provides an example of how the formula is used.

The relationship representations determined in Part A are generic. That is, they do not apply to any specific commercial or research DBMS. Part B translates the generic relationship representations into those associated with a specific database management system being designed for. Currently, the only specific DBMS under consideration is the CODASYL DBMS [DDLC78].

The rules in both Part A and Part B rules are invoked only once by the Control Module at the beginning of problem processing.

The rules for this rule group are found in Appendix C.

5.10 The Representation Restriction Rule Group

This rule group has two objectives: 1) to restrict relationship representations in a cluster implementing the design action for direct reduction of problem size (Section 4.2.2 of SP 500-151), and 2) to select relationship representations for an initial skeleton in a cluster for which selective skeleton generation has been selected.

The purpose of relationship restriction is to reduce the number of initially selected representations, identifying those non-critical relationships which may have some representations eliminated or possibly restricted to one representation having a high certainty factor (Section 4.2.2 of SP 500-151). The underlying assumption behind this design action is that for some

relationships, there is a good chance that certain initially selected representations will not yield significantly better solutions or would produce poor solutions, and therefore need not be tried. The benefit obtained by eliminating these representations is the reduction of number of alternative skeletons to examine, with correspondingly reduced problem size. The risk is that, for a particular cluster, the underlying assumption is false, and a better solution may be overlooked.

Part A contains rules which carry out relationship restriction. These rules can be invoked directly by the Control Module when this design action is selected for a specific cluster. In general, for relationships having significant workload complexity, all reasonable relationship representations are retained for further processing. For relationships without workload complexity, certain representations may be eliminated, with possible restriction to a single representation.

The rules in Part B select relationship representations for an initial skeleton for a cluster which will undergo selective skeleton generation. This skeleton will consist of the representation for each relationship in the cluster which has the highest certainty factor. These rules may be invoked many times by the Control Module for individual clusters.

The rules for this rule group are found in Appendix C.

6. DESCRIPTION OF THE CLUSTER DIVISION KNOWLEDGE BASE.

The function of this knowledge base is the selection of breakpoints from which clusters may be formed. This knowledge base contains six rule groups.

Four of the six rule groups contain rules for selecting breakpoints based on different levels of restrictiveness in the criteria used for breakpoint selection. The Cluster Division Control Rule Group of the Control Module determines which of these four rule groups will be invoked. A fifth rule group determines that certain relationships should not serve as breakpoints and that the entities in these relationships belong in the same cluster. A sixth rule group makes a final determination of the breakpoints.

Criteria for levels of breakpoint restrictiveness used by the four breakpoint selection rule groups are based on the extent of workload complexity allowed in selecting breakpoints. A high level of breakpoint restrictiveness means that only relationships with very little workload complexity can serve as breakpoints. Using lower levels of breakpoint restrictiveness means that relationships with more workload complexity can be chosen as breakpoints. This is not the same as restriction of the number of relationship representations described in Section 5.

The four breakpoint selection rule groups are ordered on the basis of the level of restrictiveness. The objective is to divide the LDS using the breakpoint selection rule group having the highest level of restrictiveness possible, thus putting the relationships having the most workload complexity inside clusters and using relationships with less workload complexity as breakpoints. The method of dividing clusters is to apply rule groups for selection of breakpoints in order of decreasing level of restrictiveness in the criteria used.

The basic structure of this knowledge base is as follows. The Cluster Division Control Rule Group determines which breakpoint selection rule group will operate. The Bond Relationship Rule Group is invoked to make conclusions that individual relationships should be within clusters and should not serve as breakpoints based on characterizations of workload complexity. Combinations of the negations of the conclusions of the Bond Relationship Rule Group together with other information are then used by the rules of the operating breakpoint selection rule group to select breakpoints. The specific types of characterizations used in the combinations of negated bond relationship conclusions constitute the differing levels of breakpoint restrictiveness of the individual breakpoint selection rule groups. The Top Level Breakpoint Selection Rule Group is then invoked for a final determination.

The rules for this knowledge base are found in Appendix D. Figure 9 shows the dependencies for the rule groups in this knowledge base.

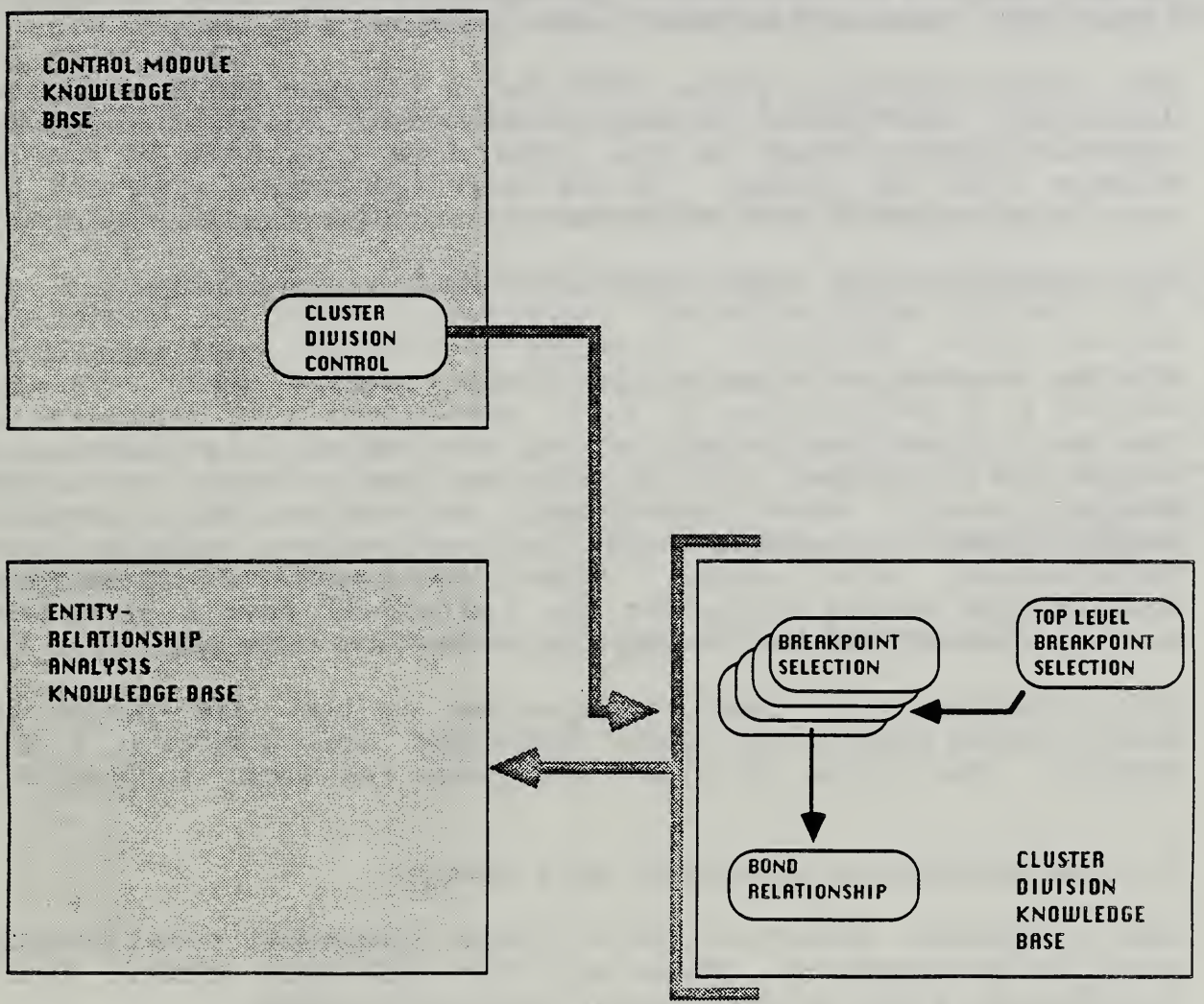


Figure 9. Dependencies Between Rule Groups In The Cluster Division Knowledge Base

6.1 The Cluster Division Control Rule Group

The Cluster Division Control Rule Group determines which breakpoint selection rule group is to be used. In general, unless the user specifies otherwise, the most restrictive group is selected. If this fails to break the LDS down to sufficient size, less restrictive groups are applied.

The rules basically state that for a cluster which will undergo division, the rule group with the level of breakpoint restrictiveness which is one level down from that of the parent cluster will be chosen. As we gain experience with the KBS, a more sophisticated set of division control rules may evolve.

The rules for this rule group are found in Appendix D.

6.2 The Bond Relationship Rule Group

The Bond Relationship Rule Group determines relationships which should be included within clusters and should not serve as breakpoints. The breakpoint selection rule groups use combinations of negations of the conclusions made by the Bond Relationship Rule Group. Specific types of these negated conclusions form a basis for the individual levels of breakpoint restrictiveness of the breakpoint selection rule groups.

This rule group is heavily dependent on the rule groups in the Entity Relationship Analysis Knowledge Base (Section 3 of this report). The rules for this rule group are found in Appendix D.

6.3 The Breakpoint Selection Rule Groups

The breakpoint selection rule groups recommend relationships to serve as breakpoints. There are four such rule groups, each with a different level of breakpoint restrictiveness.

Breakpoint Selection Rule Group 1, in Part A of Section 6.3 of Appendix D, contains the most restrictive criteria, while Breakpoint Selection Rule Group 4, found in Part D, contains the least restrictive criteria. The determination of which of these rule groups to use is made by the Cluster Division Control Rule Group (Section 6.1 of this report). Breakpoint Selection Rule Groups 2 and 3 are found in Parts B and C of Section 6.3.

The heuristics in Rule Group 1 cannot be guaranteed to subdivide an LDS enough to create good clusters. Usually, it is necessary to use breakpoint selection rules relying on weaker heuristics from less restrictive Rule Groups 2, 3 and 4. The conclusions of the Bond Relationship Rule Group (Section 6.2 of this report) are

used to form the specific conditions for these heuristics. Combinations of negations of these conclusions indicating absence of certain types of workload complexity characterizations, forwarding percentage characterizations, and absolute activity level characterizations constitute the level of restrictiveness associated with each rule group. Conclusions provided directly by rule groups in the Entity Relationship Analysis Knowledge Base are also used.

The rules for this rule group are found in Appendix D.

6.4 The Top Level Breakpoint Selection Rule Group.

The Top Level Breakpoint Selection Rule Group, makes a final selection of the breakpoint relationships based on the recommendations of the breakpoint selection rule groups in effect. The effect of this rule group is to select a relationship as a breakpoint if it is recommended by a breakpoint selection rule group and if the resulting cluster would not consist of only one entity. The primary purpose of the Top Level Breakpoint Selection Rule Group is to insure that no leaf entity relationship becomes a breakpoint, thus preventing excessive fragmentation of the LDS.

The rules for this rule group are found in Appendix D.

7. THE SKELETON GENERATION KNOWLEDGE BASE

Once a set of reasonable representations have been selected using Reasonable Representation Rule Group (Section 5.9) and once the problem LDS has been divided using the rules in the Cluster Division Knowledge Base, canonical records must be formed in each cluster (Section 4, SP 500-151).

The number of potential skeletons within each cluster varies exponentially with the number of representations for each relationships in the cluster. A particular cluster may have few or many relationships and entities. Each relationship may have several reasonable representations. Each unique combination of reasonable representations in the cluster yields a unique skeleton with a unique set of canonical records. For example, for a cluster with 10 relationships where each relationship has 3 reasonable representations, the number of possible skeletons is given by the expression:

$$3^{10}.$$

If the cluster has few possible skeletons (generally far fewer than in the example), they may be enumerated. Their workload costs can then be estimated using the cost estimation function (Section 4.5.6, SP 500-151). The canonical records from the lowest cost skeletons can then be submitted for fine-tuning (Section 4.3.1, SP 500-151).

For clusters with many skeletons, the costs of enumerating all alternatives is prohibitive. In such cases, selective generation of skeletons is more appropriate. The goal of this strategy is to use heuristics to identify a small number of low cost skeletons without generating a substantial number. The canonical records of these skeletons are then considered good candidates for fine-tuning. Sections 4.3.2 and 4.5.5 of SP 500-151 describes the process of selective skeleton generation and the conditions under which this activity is selected by the Control Module.

The Control Module controls selective generation of skeletons by invoking rule groups in the Skeleton Generation Knowledge Base (Section 4.5.5 of SP 500-151). This knowledge base has six rule groups which will be briefly described:

* The Control Module Support Rule Group for Skeleton Generation makes certain specific determinations required by the Control Module (this rule group was not described in SP 500-151).

* The Skeleton Selection Rule Group determines which skeleton is to be processed for further design activity.

* The Relationship Selection Rule Group determines which

relationships within an individual skeleton should have their representations changed.

* The Representation Selection Rule Group determines which alternative representations to use for the selected relationship. This leads to the creation of a new skeleton, for which the cost is computed using the Cost Estimation Function.

* The Skeleton Analysis Rule Group determines if the new skeleton is a candidate for further design activity or if it should be pruned from the search space.

* The Design Structure Evaluation Rule Group evaluates the cost effectiveness of each canonical record and each relationship representation in a particular skeleton and determines whether or not the record or representation should be regenerated in subsequent skeletons.

In the KBS, the Control Module invokes each of these rule groups in this knowledge base, conducting what is, in effect, a generate and test strategy. The result of using this strategy is a search for low cost skeletons and canonical records. The Control Module controls the direction of the search by invoking individual rule groups in the knowledge base. The appendix of SP 500-151 contains an example of selective skeleton generation.

The rules for this knowledge base are found in Appendix E. Figure 10 shows the dependencies for the rule groups in this knowledge base.

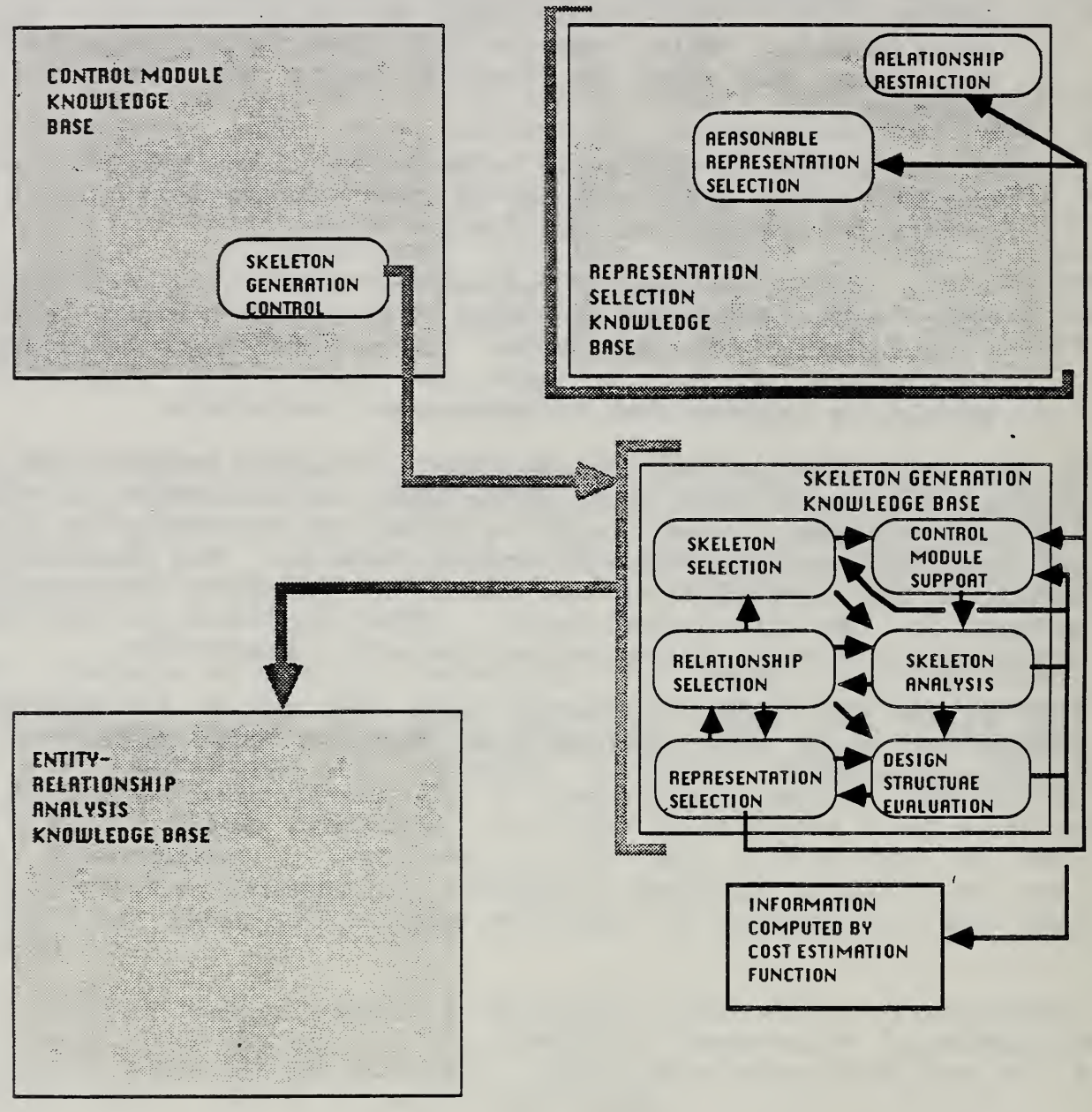


Figure 10. Dependencies Between Rule Groups In The Skeleton Generation Knowledge Base

7.1 An Important Heuristic in Skeleton Generation

Within individual clusters, the skeleton generation process is simplified by partitioning search into two main phases. In the first phase, relationship representations are varied between a many to one symbolic representation and many to one absorption, e.g. these representations are interchanged where appropriate. This is called the record phase. In the second phase, less often used representations involving combinations of symbolic and direct pointers are tried where appropriate. This is called the "relationship" phase.

There are two other phases: Limited-record and Combined. Limited-record is a more constrained version of the record phase where fewer alternatives are explored. It is used in place of record phase when a skeleton has many potential alternatives but the maximum number of alternatives that can be explored is much less. Combined phase removes the distinction between record and relationship phases, effectively creating only one phase. This is used when the potential number of alternatives to generate is very small.

7.2 The Skeleton Generation Control Rule Group (Part Of The Control Module)

The Control Module determines which task to perform and which rule group to invoke to further skeleton generation. To do this, the Control Module relies on the previous action performed, and on the state of design within the cluster. The state of design for a cluster includes the actions have thus far been performed on the cluster, the number of skeletons which have been generated, as well as other assorted factors. While these rules are actually considered to be part of the Control Module, they are included in the section of skeleton generation to make clear the context in which they are used.

The rules for the Control Module Decision Rule Group are found in Appendix E.

7.3 The Control Module Support Rule Group for Skeleton Generation

This rule group provides certain support functions to control module decision making about skeleton generation. The rule group is divided into three parts.

Part A consists of three rules for determining the number of skeletons to be generated for the cluster during skeleton generation. For clusters with a large number of skeletons, the effect of these rules is to determine an appropriate smaller

number to generate. The maximum number of skeletons to generate for any cluster is set at 100 in this rule group. It is invoked by the Control Module at the start of skeleton generation.

Part B consists of six rules for deciding whether or not at any point during the skeleton generation process to continue to generate skeletons within a cluster or to terminate activity. Skeleton generation terminates when the number of skeletons to be generated exceeds the number of skeletons to be generated as determined by the rules in Part A. The exception to this occurs when the last skeleton generated resulted in a lower cost. Part C contains six rules for determining phase, described in Section 7.1. These rules are largely based on the number of skeletons which have been generated as well as the maximum number which can be generated within a cluster. In general, record phase continues until the number of skeletons generated reaches 2/3 of the number of skeletons to be generated selectively or until all possible combinations of representations have been considered. When skeleton generation commences for a cluster, the Control Module periodically redetermines the phase until record phase ends. Once relationship phase begins, the redeterminations cease.

In fact, it may be possible that Parts B and C could be combined into a single determination. This may be done in the future.

The rules for this rule group are found in Appendix E.

7.4 The Skeleton Selection Rule Group.

The purpose of this rule group is selection of skeletons. This rule group has two parts. The function of Part A is to select a single skeleton to be worked on during skeleton generation.

The rules in Part B are invoked by the Control Module after skeleton generation is completed to determine which skeletons are to be retained for detailed design. That is, the physical records from these skeletons will undergo file organization design. The rules are based on a comparison of a skeleton's cost with that of the least costly skeleton in the cluster and the total number of skeletons to be generated within the cluster. The rules in Part B have been included in the Skeleton Selection Rule Group since the publication of SP 500-151.

The rules for this rule group are found in Appendix E.

7.5 The Relationship Selection Rule Group.

The function of this rule group is to select the next relationship which will have its representation varied within a

selected skeleton. The rule group is invoked after the Skeleton Selection Rule Group has determined which skeleton to work on. If a relationship to be varied is successfully selected, the Representation Selection Rule Group (Section 7.6 of this report) is normally invoked to determine the alternative representation to be substituted.

This Rule Group has four parts. Part A consists of rules recommending which relationships within a cluster are to have their representations varied during the record, limited-record, or combined phase (See Section 7.1 of this report).

Part B consists of RECOMMEND_FOR_ALTERATION rules which recommend relationships to vary during the relationship phase. These rules, although fewer in number, operate in the same way as the rules described in Part A. They suggest which relationships might have their representations altered to effect changes in the types of pointers which are used. Basic canonical record structures are not effected.

The rules in both Part A and Part B rely on several important functions for selecting relationships and determining criticality. These functions represent heuristic information based on itemized workload costs for individual relationships and entities provided by the Cost Estimation Function (Section 4.5.6 of SP 500-151). They independently access the database to obtain these costs. Functions are used instead of rules for the sake of computational efficiency. Five functions select relationships to vary:

MOST-COSTLY-REL - provides recommendations for relationships not currently represented by absorption which have high itemized costs for workload traversing the relationship;

REL-ROOT-ACC-COST - identifies a root absorbing entity in a complex physical record which has a high cost for direct large subset retrieval; it then recommends the relationship in the record represented by absorption which has the smallest total frequency;

ABSORB-ENT-ACC-COST - recommends a relationship represented by absorption for which the absorbed entity has a high cost of direct access;

COMPLEX-ABS-REL - selects record structures having more than one entity whose proportion of the total workload cost of the entire skeleton exceeds its proportion of the total workload frequency of the skeleton; it then selects the relationship in the record represented by absorption with the smallest frequency of traversal.

OVERSIZE-REC - if device length restrictions have been specified, identifies canonical records which exceed a device length and recommends a relationship along which to break the record.

The function *COMPUTE-CRITICALITY* is used to determine criticality, which ranges from 0.0 to 1.0. Relationships having the highest criticality are varied first. The criticality of an individual relationship is computed by adding a constant contained in the rule (currently set at 0.5 in all rules) to the percentage of the relationship's total cost in the skeleton. Criticalities higher than 1.0 may be provided directly by other rules.

Part C consists of rules which recommend that relationships should not be varied. These DO_NOT_VARY rules are of two types: INITIAL and SUBSEQUENT. INITIAL DO_NOT_VARY rules fire when individual skeletons are first generated and evaluated by the Cost Estimation Function. SUBSEQUENT rules fire as work on an individual skeleton progresses. Just as for RECOMMEND_FOR_ALTERATION rules, more than one DO_NOT_VARY rule may fire for an individual relationship and more than one relationship may be recommended for alternation.

DO_NOT_VARY conclusions are made for two major reasons: 1) a relationship may not have a new representation which can be chosen (e.g. all may have been tried), or 2) an efficient representation has been identified for the relationship.

Finally, Part D selects which relationship, among the several which may be recommended, should be altered. In making this selection, these rules (only three in number) combine the conclusions of RECOMMEND_FOR_ALTERATION and DO_NOT_VARY rules. These rules rely on total criticality levels to order relationships and to select the relationship with the highest criticality.

The rules for this rule group are found in Appendix E.

7.6 The Representation Selection Rule Group.

The function of this rule group is to choose alternative representations for a relationship which has been selected for alteration by the Relationship Selection Rule Group. It is invoked by the Control Module for this purpose after the relationship to alter has been chosen. Individual rules concluding RECOMMEND_NEW_REPRESENTATION fact expressions may fire to recommend alternative representations for a chosen relationship, each with an associated certainty factor. It is possible that more than one rule may fire, resulting in the recommendation of more than one relationship representation.

Most of the rules concluding RECOMMEND_NEW_REPRESENTATION fact expressions are used during the record phase. Only two rules apply in the relationship phase.

The basic pattern of each rule concluding a RECOMMEND_NEW_REPRESENTATION fact expression is to recommend an alternative representation for a chosen relationship if the following conditions hold: the alternative representation is among the initially selected reasonable representations with a confidence factor above some threshold (currently set very low), the representation is consistent with the current phase, the representation has not been found to be inefficient in previously generated skeletons, and the resulting changes would not produce a duplicate skeleton. The initial set of reasonable representations is determined at the start of overall problem processing by rule groups in the Representation Selection Knowledge Base (see Section 4 of this report). Determinations of inefficient relationship representations are made by the Design Structure Analysis Rule Group (described in section 7.8 of this report).

Rules concluding RECOMMEND_NEW_REPRESENTATION fact expressions may recommend alternative representations for a single chosen relationship. However, they may also recommend alteration of two different relationship representations at once. They may recommend alternatives to the chosen relationship and to a second relationship having a common entity with the chosen relationship. In this case, a second relationship has its representation altered along with the originally chosen relationship. The effect of this is to vary two relationships at once, resulting in a more fluid alteration of record structures. The rules for this rule group are found in Appendix E.

7.7 The Skeleton Analysis Rule Group.

The purpose of this rule group is to analyze newly generated skeletons during the skeleton generation process. The rule group currently performs two valuable functions: 1) to identify inefficient or infeasible skeletons for elimination from further processing, thus permitting the search space to be pruned, and 2) to identify skeletons which constitute special cases or which have special characteristics. Other analysis functions within the scope of this rule group may be added in the future.

The rules for this rule group are found in Appendix E.

7.8 The Design Structure Evaluation Rule Group.

This rule group currently consists of only five rules. Its function is to identify individual relationship representations

and record structures which are particularly efficient or inefficient. That is, these rules address specific representations for individual relationships in the context of specific canonical record structures that the representations, in part, define. Inefficient structures are not regenerated during the skeleton generation. Efficient structures may be regenerated if appropriate. This rule group is used by the Representation Selection Rule Group and the Skeleton Analysis Rule Group.

The rules for this rule group are found in Appendix E.

8. CONCLUDING REMARKS

This report has described the knowledge bases and rule groups contained in the Knowledge-Based System For Physical Database Design.

Development of individual knowledge bases is continuing, and is aimed at increasing the system's capability to do design of databases using commercial database management systems. It is expected that testing the system on real world problems will contribute significantly to the development and refinement of rules for database design.

The long term goal of this project is to produce an in-house knowledge-based system capable of doing physical database design in a wide variety of hardware and software environments. In doing this, we hope to learn more about how to do physical database design and about the capabilities and uses of knowledge-based systems.

9. REFERENCES

[ANSI86] American National Standards Institute, Inc., American National Standard, Database Language SQL, ANSI X3.135-1986, New York, New York.

[AUER81] "Efficient File Organization Design", Report No. 23-01-05, Auerbach Publishers Inc., Pennsauken, New Jersey, 1981.

[CARL80] Carlis, John V., "An Investigation into the Modeling and Design of Large, Logically Complex, Multi-user Databases," Ph. D. thesis submitted to University of Minnesota, Minneapolis, Minnesota 55455, December 1980.

[CHAR80] Charniak, Eugene, Riesback, C. K., and McDermott, D. V., Artificial Intelligence Programming, Lawrence Erlbaum Associates, Hillsdale NJ, 1980.

[CHEN76] Chen, P. P., "The Entity-Relationship Model - Toward a Unified View of Data," ACM Transactions on Database Systems, March 1976, pp. 9-36.

[CLOC84] Clocksin, W. F., and Mellish, C. S., Programming in Prolog, Springer-Verlag, New York, 1984.

[CUGI87] Cugini, John V., Programming Languages For Knowledge-Based Systems, NBS Special Publication 500-145, National Bureau of Standards, February, 1987.

[DABR88] Dabrowski, C. E. and Jefferson, D. K., A Knowledge-Based System for Physical Database Design, NBS Special Publication 500-151, National Bureau of Standards, February, 1988.

[DDLC78] Data Definition Language Committee, "CODASYL Data Definition Language Committee," Journal of Development 1978, Canadian Government Publishing Centre, Ottawa, Ontario, K1A0S9.

[MARC78] March, Salvatore T., Jr., "Models of Storage Structures and the Design of Database Records Based Upon a User Characterization," Ph.D. thesis submitted to Cornell University, May 1978.

[NILS80] Nilsson, Nils J., Principles of Artificial Intelligence, Tioga Publishing Co., Palo Alto, CA, 1980.

[SHAF76] Shafer, G., A Mathematical Theory of Evidence, Princeton University Press, Princeton, 1976.

[STOR88] Storey, Veda C., and Goldstein, Robert C., "Expert Systems for Automation of Database Design", Submitted for publication, 1988.

[TEOR82] Teorey, Toby J., and Fry, James P., Design Of Database Structures, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1982.

[THOM85] Thompson, Terence R., "Parallel Formulation of Evidential Reasoning Theories," Proceedings of the Ninth International Joint Conference on Artificial Intelligence, Los Angeles, CA, 1985.

NBS-114A (REV. 2-8C)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET (See instructions)	1. PUBLICATION OR REPORT NO. NISTIR 89-4139	2. Performing Organ. Report No.	3. Publication Date August, 1989

4. TITLE AND SUBTITLE

A Detailed Description of the Knowledge-Based System for Physical Database Design

5. AUTHOR(S)

6. PERFORMING ORGANIZATION (If joint or other than NBS, see instructions)

NATIONAL BUREAU OF STANDARDS
U.S. DEPARTMENT OF COMMERCE
GAITHERSBURG, MD 20899

7. Contract/Grant No.

8. Type of Report & Period Covered
Internal Report (IR)
7/85 - 12/88

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (Street, City, State, ZIP)

10. SUPPLEMENTARY NOTES

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)

A knowledge-based system for physical database design has been developed at the National Computer Systems Laboratory. This system was previously described in NIST Special Publication 500-151. This is a follow-up report to that publication which describes the knowledge base for this system in detail. The description includes a complete explanation of each component of the knowledge base together with the actual rules used by the system.

12. KEY WORDS (Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)

certainty factor; entity-relationship model; inference engine; knowledge-based system; logical data structure; physical database design.

13. AVAILABILITY

☒ Unlimited

☐ For Official Distribution. Do Not Release to NTIS

☐ Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.

☐ Order From National Technical Information Service (NTIS), Springfield, VA. 22161

14. NO. OF PRINTED PAGES
62

15. Price
$14.95